Vicariously Sharing Captured Web Experiences through an Automated Recommendation System

Khai N. Truong, Gregory D. Abowd
College of Computing & GVU Center
Georgia Institute of Technology
801 Atlantic Drive
Atlanta, GA 30332-0280 USA
+1 404 894 7512
{khai, abowd}@cc.gatech.edu

Maria da Graça Pimentel
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo
Caixa Postal 668
São Carlos, SP 13560-0960 BR
+55 16 273 9657
mgp@icmc.sc.usp.br

ABSTRACT
Our daily experiences are rich in content that we want to recall in the future. These previous experiences are often where we begin in our search for information needed when we work. In a community, we can rely on others to suggest materials when our own expertise fails to provide us with what we need. Likewise, others will make referrals only from things that they have previously experienced. In this paper, we present WebMemex, a system that recommends related Web pages to what the user is currently viewing. This system acts as an instantiation of an architecture to automatically capture and access information in a manner similar to when a person is searching for information related to her current work context—where the related information being retrieved is something she has previously seen or that her friends have seen before and could ultimately suggest to her.

Keywords
Ubiquitous computing, automated capture and access applications, information seeking, recommendation systems, asynchronous collaboration.

INTRODUCTION
Information we have seen in the past can often be useful in two specific ways:

1. It can be directly useful to us when we need it at a later time;
2. It can be useful to others if we can share it with them at times when they need it.

Knowing that information will be useful in the future is not easily predictable. As a result, when we want to retrieve previously view information, often we find that we had not noted (mentally or physically) enough about the relevant pieces of information for recall.

One of the themes of ubiquitous computing is the automated capture of everyday experiences made available for future access. Automated capture and access applications leverage what computers do best – record information. In return, humans are free to fully engage in the activity and to synthesize the experience, without having to worry about tediously exerting effort to preserve specific details for later perusal. In this paper, we present an application known as WebMemex. WebMemex is an automated capture and access application built in the same spirit of Vannevar Bush's *memex*, but geared towards recording URLs that have been visited in the past [2].

While Web browsers do have existing history mechanisms, they are impoverished. Too often when we want to retrieve previously viewed Web information, we have either forgotten to bookmark the relevant URL or we cannot use the history mechanism stored on the browser machines. The existence of bookmarks and browsing histories tied to specific browser machines makes it less useful to mobile users.

In the WebMemex system, the user's web surfing activity is continuously recorded and used to automatically recommend relevant Web pages to the user that she has previously seen. The captured information is also shared with other people the user knows to suggest related information during their Web experience.

Overview of the Paper
We begin by providing the complete motivation behind capturing and sharing of user's Web surfing history. We then present the WebMemex prototype and how a user can use the system. We will explain the WebMemex architecture and how the system is constructed.

As we prototyped this system, we encountered many different issues related to the specific challenges of capture, access and sharing of Web content. However, our solution for capture and access of Web content is

generalizable to other captured experiences. We will discuss the decisions made in the WebMemex system and address more general questions uncovered, as well. A review of related work will discuss how our approach is different from traditional recommendation systems and other existing capture and access applications, as well. We conclude with some discussion of future directions of this work.

MOTIVATION

The effort of finding information over the Web has been greatly facilitated over the recent years with great improvements in the quality of the results returned to us by search engines (such as Google.com and Altavista.com) and recommendation systems (such as Amazon.com and Reel.com). Despite these improvements, users still perceive friends as the best source of good and useful recommendations and have very high trust in the recommended information [16]. Whereas the algorithms used by recommendation systems are not usually intuitive to the users, it can be assumed that friends have common understandings and interests.

In Figure 1, we present an instant messaging dialog between two "buddies". In this exchange, one user asks a second to recommend information about video display tablets that she might have seen in the past. The second user suggests a particular piece of hardware that she is able to recall. This piece of information also triggers the second user to remember about an additional relevant Web page, as well.

This simple scenario highlights the following important points:

- People look to their friends for suggestions when a problem extends beyond their own expertise;
- People retrieve things they have previously seen for answers; and
- URLs we have seen in past Web experiences are useful information that we may want to retrieve in the future.

An additional point in this example is the non-traditional use of instant messaging. Studies of instant messaging include social awareness, bonding, and activity organization and scheduling [12]. Instant messaging also lends itself to the quick sharing of (or request for) information, such as URLs. As Grudin pointed out, groupware is successful when there is a clear benefit to using it and not much effort is required in return [5]. With instant messaging, when a friend is available and has the expertise that we need, asking her for help in finding information is often easier than finding it ourselves.

Finding the friend that might be able to recommend information to us, as we need it, can be a challenge. Knowing which person among the group of friends to ask is the first issue. And even when we know the exact person that can provide us with the suggestions, at best, her availability will be variable at the moment we need to ask her for information.

In this work, we present a system that continuously captures each user's Web experiences and makes it available to her friends to provide asynchronous recommendations. Users can vicariously make recommendations to their friends without 1) needing to be present and 2) giving information to those who do not want it. Rather than inappropriately spamming a list of people (many of whom may not be interested), when the user comes across something interesting, the relevant URL and its Web page contents are captured and will be automatically recommended when people develop an interest in it (or something similar).

In a scenario where the user is using WebMemex, she would not need to find her friends to ask for their recommendations about video display tablets they have seen in the past. Instead, with the Web experiences of the

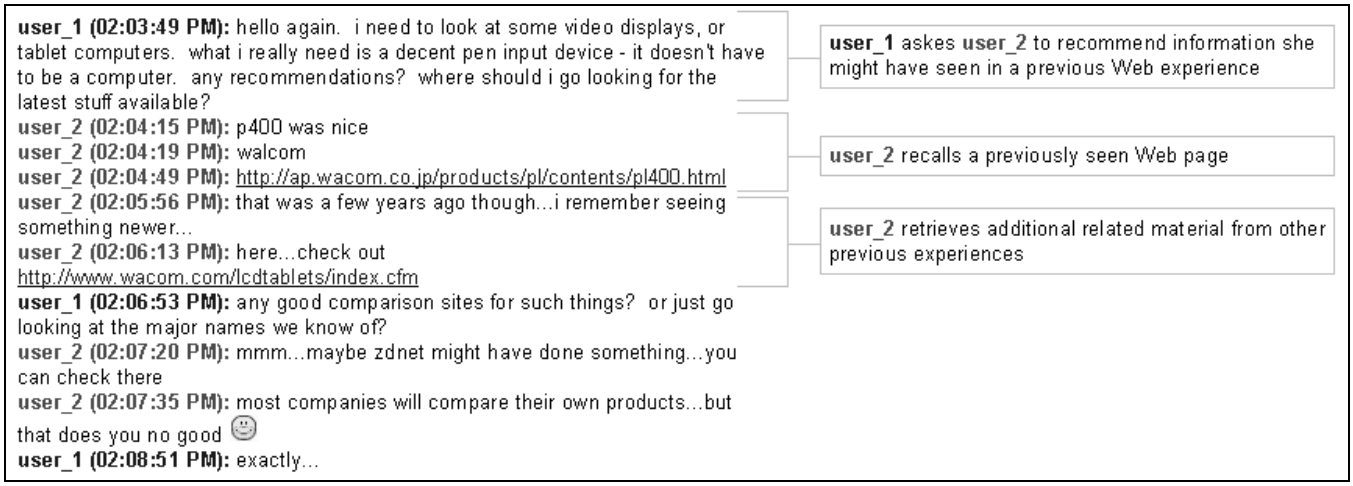

Figure 1. An instant messaging exchange where one user recommends to a second user relevant Web pages to view.

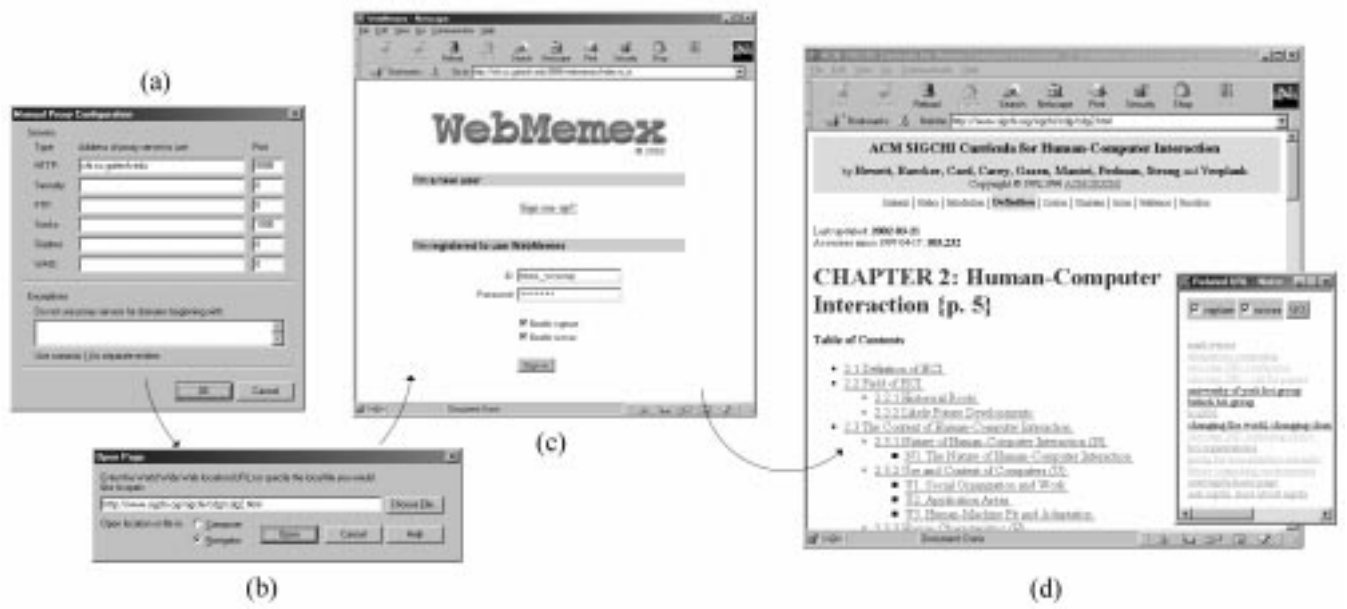

Figure 2. The WebMemex Prototype. To use WebMemex, a user simply configures of a browser to talk to a proxy server that captures Web histories (2a). When a user begins a Web surfing session (2b), the browser automatically goes to a screen for the user to sign (2c). Once authenticated, the Web browser is fully augmented to capture and suggest URLs from the Web surfing history of the user and her friends. The related URLs are shown in a small window outside of the Web browser (2d).

user's friends captured, when she begins her search for this device, the system recognizes that previously seen Web pages about "video display tablets" would be of interest to her. The system will then retrieve the list of Web pages that her friends have viewed on this matter; if any were found, they are recommended to the user as suggested related materials.

The recommendation capability provided in this system is different from those of traditional recommendation systems, from which a user expects to ask the question "can you recommend me the best X for Y?" This work explores the suggestion of materials that addresses the question of "can you recommend me some X for Y that you know about?" More emphasis is placed is the investigation of how to support the capture and asynchronous sharing of information between groups of people.

THE WEBMEMEX PROTOTYPE

In his 1945 Atlantic Monthly article, Vannevar Bush described his vision of the *memex*, a generalized capture and access application [2]. The memex is a system intended to store the artifacts that we come in contact with in our everyday lives and the associations that we create between them. He noted that a "record … must be continuously extended, it must be stored, and above all it must be consulted."

WebMemex is system built in the same spirit as Bush's vision of the *memex*, but geared towards records of Web visits. To create WebMemex, we augmented a standard Web browser with a simple capture capability that serves as a new history feature. This enhanced Web history can be used to support a number of access features. One access feature supported, though not discussed in this paper, is the ability to perform searches over personal Web histories, allowing the user to revisit her previous navigation trails [21]. In this paper, we demonstrate a recommendation capability (an asynchronous collaborative access feature), where the system uses previous Web experiences to enhance a user's Web session with suggestions of related URLs (see Figure 2d).

Capturing Web Pages Continuously

To capture a user's web surfing history, we needed to be able to monitor the web pages she visits. We initially explored the implementation of hooks or listeners for common Web browser's such as Internet Explorer®; however, we found this method was not ideal because client-based solutions are too platform-specific.

Unfortunately, a user typically works on more than one machine. However, all Web history should be accessible from any networked machine, regardless of the machine on which the URL was initially visited. Thus, as a user works on different machines, the Web pages she visits need to be integrated with those previously captured (as a continuous Web experience).

Hence, we leverage on existing Web browser's ability to talk to an HTTP proxy. On existing Web browsers, the user can quickly specify the location where the proxy server is running, as in Figure 2a. When the user begins her Web surfing, the Web browser will talk to the Web proxy server. This proxy server initially checks to see if it knows who the user is (i.e., if the user is registered and logged into the system). This step allows the user to be able to log into different machines using the same ID as she works on different machines (Figure 2c); but the information will automatically be tied together with her previously captured information.

Obtaining Suggestions

In addition to handling HTTP requests and delivering Web content back to the browser, the Web proxy logs and reacts only to a specific document type. When the content type `text/html` is served back to the Web browser, the system appends some Javascript at the end of the document to invoke the opening of the small pop-up window that shows related Web pages. This Javascript in no way affects the rest of the content on the page.

When a user views a Web page, the small pop-up window is informed of the current URL being viewed and it is responsible for obtaining the list of related URLs. In WebMemex, a simple method was used to tie together what are obviously related Web visits that a person and her friends might have come across. Keywords are used to determine if Web pages are related to one another, but can be replaced by more sophisticated techniques if it is not found to be adequate. The most relevant documents are those documents matching the most number of similar keywords as the current Web page the user is viewing. Because users have personal experience with information they have seen in the past, suggestions coming from her own experience has higher relevance scores than those from others. Using these two factors, we compute each recommendation's relevance score.

Figure 3. Suggested Web pages color coded based on computed & quantized relevance scores.

The relevance scores are then used to determine how each suggestion should be presented to the user. In suggesting information, an interface should encourage the user to explore the options. The challenge is to allow the user to quickly determine the most relevant Web pages to what she is currently looking at, but not necessarily spatially bias the information by ordering the information. Instead, suggestions are color coded to allow the user to quickly visually determine the suggested page that may have the most relevance (see Figure 3). To make this determination intuitive to the users, we use only varying grayscale colors. Because it is difficult for the human eye to easily perceive the difference between small changes in color, results are quantized. The quarter of the results with the highest relevance scores are the darkest and the bottom quarter of the results are in light gray.

Controlling Capture & Access

In some situations, the user may not want to have her Web history preserved; or the user may view a page that she does not want to share with others. At other times, recommendations are not needed. The interface gives the user two places to specify the services she wants. At sign-in time, the user can tell the system to "enable capture" and "enable access" (see Figure 3c). The default is that both capture and access are enabled. During the course of the surfing session, it is also possible for the user to change their mind about whether or not to have the content captured and shared or not. At anytime during the session, the user can change this setting by specifying this on the small pop-up WebMemex window (see Figure 3d).

The availability of these controls in the WebMemex window also allows the user to retroactively determine this effect at the page level. When visiting a page, if the user decides that it should not be captured, she can unselect the capture option. The user can continue surfing without having her navigation captured until the capture option becomes selected again. This option essentially is a way for the user to say, "do not mark that I have visited the page I am viewing right now." If the user has viewed the page in the past, it does not remove these visits from the history. While privacy may lead to the desire for being able to modify retroactively the capture history of more than just the currently viewed page, it is not completely necessary for the user to manually specify that she did not view a page. This same effect is achieved by protecting anonymity of the source of the suggested materials.

Initially, we had separate options for a user to specify whether she wants to share what she captures with her friends or not. And likewise, we had a separate option for whether or not she wants to receive recommendations that include the experiences of her friends or just her own. These options were removed in favor of protecting 1) privacy, and 2) the spirit of sharing. Allowing a user to specify whether or not to include the recommendations

coming from her friends' experiences or not means being able to decide deductively if a particular recommendation is coming from her set of experiences or others. To preserve the spirit of sharing, the option to capture information and the option to share what is captured were reduced to a single option to enable capture. Thus, all information that is captured is shared. We will discuss how the issue of privacy is further preserved in later in the paper.

HIGH-LEVEL ARCHITECTURE

The system itself was built using an infrastructure for capture and access application, known as INCA [19]. We will not discuss INCA itself in this paper, but will explain how WebMemex was constructed from a high-level architecture and describe how information is stored, retrieved and presented to the users.

As mentioned before, the WebMemex service is provided through an augmented Web proxy server. As a user requests a Web page, the Web proxy server retrieves the request and serves it to the registered user on the requesting client browser. If the user has capture and access services enabled, the Web proxy server will react appropriately on the information being returned to the user. We will now describe how the system is designed to support these features.

Proxy, Capture & Share

The suggestion of related material can be considered an added feature to the Web surfing activity. However, it should not get in the way of the user navigating through the Web; i.e., the capturing and sharing of information should occur without slowing down Web visits. As a result, the WebMemex server separately handles the concerns involved; see Figure 4 for the architectural diagram. When information is requested, the proxy retrieves the information and immediately delivers it back to the browser. If capture is enabled by the user, then the document retrieved and forwarded to the browser is passed to a capture component. Likewise, the access component is informed about the user's desire for recommended material and it queries for the related material and writes the suggestions back to the browser.

As mentioned before, the proxy only logs information returned to the Web browser when the content type is `text/html`. `HTML` documents can be processed for additional understanding of what they are about, such as their title or keywords, etc. As a result, when an `HTML` document is served to the Web browser, the capture component of the proxy server captures the visit by tagging it with the URL, the title, up to 10 keywords for that Web page, the time that Web page was visited, the IP address of the browser machine, and the user's ID.

The process of keyword extraction we employed consisted of two steps. If the `HTML` document already contained meta-tags specifying keywords, we used those as the keywords for that page. Otherwise, an occurrence counter of all the words in the `HTML` body (with `HTML` tags

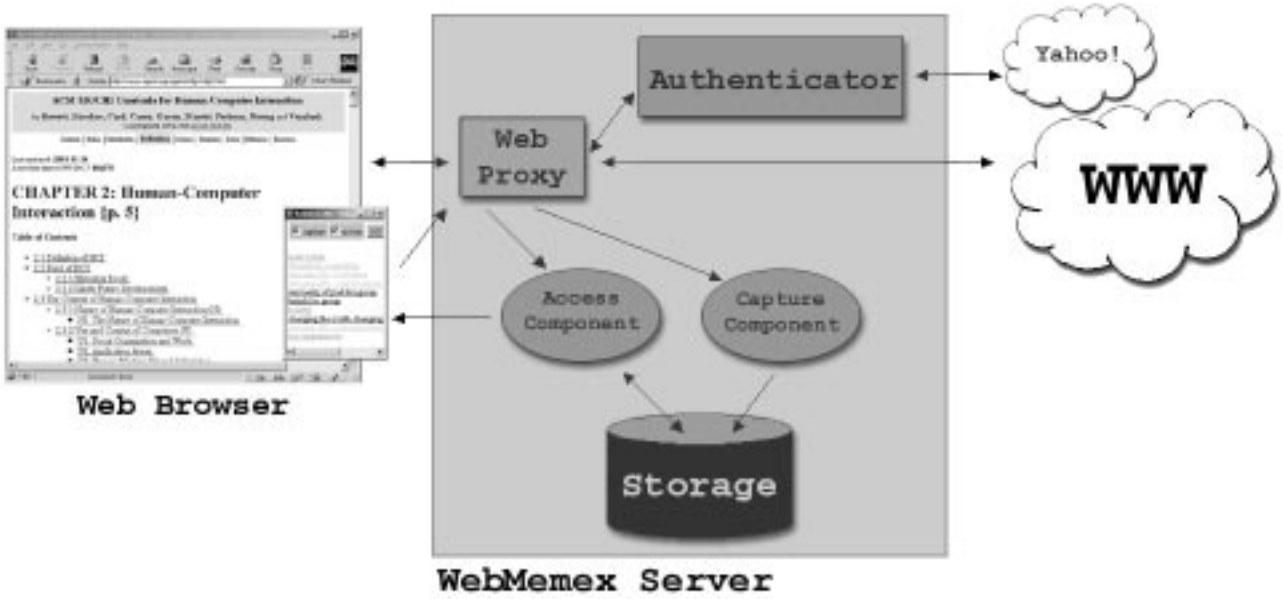

Figure 4. WebMemex Architecture. Web browsers make HTTP requests that get served by a Web proxy server after the user has been authenticated with a valid Yahoo! user ID and password. After this point, the proxy server retrieves content from the WWW for all requests and return it to the browser. If capture is enabled, a capture component is responsible for storing the user's Web visit. Likewise if access is enabled, an access component retrieves related URLs to the currently viewed Web page and writes the suggestions back to the small WebMemex window.

stripped out) and a stop list were used, where up to the top ten most frequent words are used as keywords.

The access component uses the keywords for the HTML document the user is viewing and retrieves from the storage repository all the pages that match on keywords that were also tagged with the Web surfer's signed-in user ID or the IDs of her friends. The URLs are then returned to the small pop-up browser window. We discussed earlier in the paper how the information is presented to the user.

Identifying the User & Her Friends

In talking to a Web proxy server, the Web browser must establish a connection to the server for each HTTP request. The Web proxy server maintains a list of IP addresses it has heard from, the user ID for that IP address (if the user has successfully logged in or null) and the last time there was activity from that machine. If a user is idle for more than an extended period of time the user must sign in again. While the user ID is null, the Web proxy server will only deliver back to the browser the sign-in page or the help page until the user has successfully authenticated.

Rather than maintaining a set of user IDs and passwords, the WebMemex system unloads this responsibility to another system. To use WebMemex, a user must register for a "Yahoo! Messenger" account. The instant messaging application, itself, does not need to be running or even installed; however, a user must have an account with Yahoo! Because the Yahoo! Messenger system can verify correct ID and password combinations for valid Yahoo! accounts, when a user signs in to use WebMemex, this information is simply confirmed with Yahoo!

The decision to leverage Yahoo! accounts goes beyond the simple ID and password verification service. Instant messaging applications typically are used for synchronous communication between friends. A user's online circle of friends is stored in the user's account as a buddy list. This "buddy list" allows the WebMemex system to determine with whom a user's captured history may be shared.

Rather than needing to maintain our own user authentication system we used Yahoo! Messenger's system. This gave us the added bonus of obtaining a buddy list for each user. As described above, this buddy list determines from whom suggestions for related materials should come as a person is surfing the Web.

Protocol for Asynchronous Collaboration & Automated Recommendations

Use social conventions

It is possible that some of the people whom a user has on her buddy list do not have her on theirs. As a result, the sharing of a user's captured Web history should not be based on just a single buddy list. There may be people who do not want to share captured histories with the user.

In normal social conventions, when a person has a question, she would ask people she considers her friends for advice. If the person asked considers her to be a friend, a response is probably returned. This is not to say that if the person asked does not consider her to be a friend, a response will not be returned. However, in such scenarios, how much trust to put into the response is questionable. To avoid this problem, the protocol to WebMemex uses to support the asynchronous collaboration is to share information only between reciprocating friends. With the list of people the user considers her friends, the system checks for the subset that has her also on their buddy list. WebMemex uses this subset of reciprocating friends to check their captured histories for related Web pages to suggest back to the user. We use the term *reciprocating* friends when referring to two people who both consider each other friends.

Share only with current list of reciprocating friends

This design decision also helps to resolve a second problem that arises in the domain of information sharing. Because a person's social circle often changes, whom a person considers as friends one week can be dropped from her buddy list the next week. Likewise, a new friend may develop a need for some information a user saw some time before they became friends. In such scenarios, should an old friend still be able to see information that she could have seen when she was friends with the user, and should the new friend be allowed to see something a user captured some time ago?

Each time the user signs in, the system verifies the user's ID and password with Yahoo! A matching ID and password allows the system to retrieve the user's buddy list. The buddy list is cached rather than keeping the password (for obvious security reasons and because the user might change the password) to constantly retrieve this information. The cached buddy list acts as what the system knows as the user's last known set of friends the user is 1) willing to ask for information, and 2) with whom she is willing to share information. Using this cached list of buddies, it is possible to determine the set of reciprocating friendship for every user.

This real-time method for determining reciprocating friends resolves the issue of changing group dynamics. When a user is surfing the Web, she will only receive recommendations from the people who are currently her reciprocating friends.

Protect Privacy

When a user does not have more than one reciprocating friend, suggestions from others are not included in what is returned to the user. This protects privacy because a user can easily determine that if she knowingly hasn't visit a Web page that was suggested and has only one friend on her buddy list; then obviously the friend must have visited the page.

HANDLING PRIVACY

The idea of continuously capturing and sharing information means that privacy is an important issue that must be addressed. As a result, throughout the paper, we have mentioned a variety of ways privacy is protected as we discussed different things. Decisions such as not including recommendations from others when a user does not have more than one reciprocating friend and only making information available to who is currently on a person's buddy list are examples of ways privacy is protected in the system. The constant availability of a way to quickly turn capture on and off allows the user to determine when information becomes too personal during a session and when it is something she is willing to share.

Another common way for people to protect their privacy online is to have multiple user profiles. A person can choose to have many different profiles for different situations (such as work or play) if she wants. In this scheme, a user can create a profile where she does not have anyone on her buddy list if she wants the constant capture and the recommendations, but does not want to share information with anyone. Likewise if a user has different people she shares different kinds of information with, she can use a different profile

RELATED WORK

Existing research in collaborative filtering systems —such as GroupLens [8], Ringo [15], and Movie Lens [3]— do user profiling and apply a collection of algorithms such as traditional data mining, nearest-neighbor collaborative filtering, and dimensionality reduction to cluster relevant information for recommendations. Popular Web sites such as Amazon.com, MovieFinder.com, CDNow.com and Launch.com have placed collaborative filtering technology into authentic use settings, demonstrating have proven to be accurate enough in the specific entertainment domains. However, the success of these systems has not been met in other domains or more general experiences because collaborative filters compute predictive models based on heuristic approximations of human processes.

Rather than trying to predict user profiles and creating social clusters based on dimensions of interest, our system will use what the user defines as her social circle. As a result, recommended information is coming from a well-known list of people to the user, people with whom the user has common ground and interest and most importantly with people whom she trusts. Each Web page the user visits is captured and compared against all other Web pages she has previously seen and those that her friends have seen. From this point, traditional data clustering methods can be used.

There have been many different capture and access applications that have investigated the preserving of experiences in the classroom [1, 11], meetings [10, 14, 18], and other general domains [4, 6, 17] for later perusal. For a detailed review of existing capture and access applications, review [20]. There is a lack of exploration of applications that performs the access of information during capture. As a system that recommends related Web pages to what the user is currently viewing, WebMemex continually accesses captured data as capture occurs.

In the domain of recommending related materials as the user surfs the Web, many applications (such as Letitizia [9], WebWatcher [7], and Margin Notes [13]) rely on local context or a short-term user profile and functions in a manner similar to the Remembrance Agent [13]. In a different approach, we are investigating the recommending of related material coming from a long-term capture history authored by the user as she interacts with the Web application.

CONCLUSION & FUTURE WORK

Motivated by Bush's article As We May Think, Douglas Engelbart envisioned the use of computer-based tools to augment human intellect and improve our overall ability to tackle the problems. In his work at the Bootstrap Institute, Engelbart coined the term "Collective IQ" to describe how a group can "leverage its collective memory, perception, planning, reasoning, foresight, and experience into applicable knowledge" to solve the problems of the users.

We have built an a capture and access application to explore the visions of Vannevar Bush and Douglas Engelbart. Our WebMemex continuously captures users' web surfing history and uses this history to provide the user and her friends with suggestions of related Web pages to the one they are currently viewing. This system acts as an instantiation of an architecture for capturing and asynchronously sharing experiences for the automated recommendation of related information.

The WebMemex application is currently being deployed; and we will study how an automated capture and access Web application is adopted by users. As users are allowed to create many profiles and they can chose to use this application in different ways:

- as a personal application which only recommends information from her own personal Web history; or
- as a collaborative application where information is shared between groups of friends and recommends related URLs from the collective Web history.

We will study how it is used and examine if the system is found to be useful for individual users and/or a group of users.

ACKNOWLEDGMENTS

This work was funded by a joint grant between National Science Foundation and CNPq in Brazil (U.S grant 0070345)

REFERENCES

1. Abowd, G.D., *Classroom 2000: An experiment with the instrumentation of a living educational environment*. IBM Systems Journal, 1999. **38**(4): p. 508-530.
2. Bush, V., *As We May Think*, in *Atlantic Monthly*. 1945.
3. Dahlen, B.J., et al., 1998. *Jump-starting movielens: User benefits of starting a collaborative filtering system with "dead data"*. University of Minnesota TR 98-017.
4. Deitz, P. and W. Yerazunis. *Real-Time Audio Buffering for Telephone Applications*. In the proceedings of *UIST'01* Orlando, Florida.
5. Grudin, J., *Groupware and Social Dynamics: Eight Challenges for Developers*, in *Communications of the ACM*. 1994. p. 92-105.
6. Hindus, D. and C. Schmandt. *Ubiquitous Audio: Capturing Spontaneous Collaboration*. In the proceedings of *Computer Supported Collaborative Work 1992* Toronto, Canada.
7. Joachims, T., D. Freitag, and T. Mitchell, *"WebWatcher: A Tour Guide for the World Wide Web"*. Fifteenth International Joint Conference on Artificial Intelligence (IJCAI'97), 1997.
8. Konstan, J., et al., *GroupLens: Applying collaborative filtering to Usenet news.* Communications of the ACM, 1997. **40**(3): p. 77-87.
9. Lieberman, H. *"Autonomous Interface Agents"*. In the proceedings of *CHI'97*.
10. Moran, T.P., et al. *"I'll Get That off the Audio": A Case Study of Salvaging Multimedia Meeting Records*. In the proceedings of *CHI 1997* Atlanta, GA.
11. Mukhopadhyay, S. and B. Smith. *Passive Capture and Structuring of Lectures*. In the proceedings of *ACM Multimedia 1999* Orlando, FL.
12. Nardi, B., S. Whittaker, and E. Bradner. *Interaction and Outeraction: Instant Messaging in Action*. In the proceedings of *CSCW 2000* Philadelphia, PA.
13. Rhodes, B.J., *Just-In-Time Information Retrieval*, in *Media Laboratory*. 2000, MIT.
14. Richter, H., et al. *Integrating Meeting Capture within a Collaborative Team Environment*. In the proceedings of *UbiComp 2001* Atlanta, GA.
15. Shardanand, U. and P. Maes. *Social Information Filtering: Algorithms for automating "Word of Mouth"*. In the proceedings of *CHI 1995* Denver, CO.
16. Sinha, R. and K. Swearingen. *Comparing Human Recommenders to Online Systems*. In the proceedings of *Delos-NSF Workshop on "Personalisation and Recommender Systems in Digital Libraries"*.
17. Stifelman, L.J., *The Audio Notebook*, in *Media Laboratory*. 1997, MIT.
18. Streitz, N.A., et al. *DOLPHIN: Integrated Meeting Support across Liveboards, Local and Remote Desktop Environments*. In the proceedings of *Computer Supported Collaborative Work 1994* Chapel Hill, NC.
19. Truong, K.N. and G.D. Abowd. *Architectural Support for Building Automated Capture & Access Applications*. Submitted to *UIST 2002*.
20. Truong, K.N., G.D. Abowd, and J.A. Brotherton. *Who, What, When, Where, How: Design Issues of Capture & Access Applications*. In the proceedings of *UBICOMP 2001* Atlanta, GA.
21. Truong, K.N., et al. *Implicit and explicit access of captured media trails*. Submitted to *ICME 2002*.